An Introduction to R

Introduction

R is a statistical computing package which will run under most commonly used operating systems. It will be used in all statistics modules taught by the School of Mathematics at the University of Leeds and is installed on the networked PCs around the campus. If you have your own PC, you can download your own copy of R from the comprehensive R network website: cran.r-project.org/. Some guidance on the installation procedure is given at the end of this document.

A much longer *Introduction to* R is available from cran.r-project.org/doc/manuals/R-intro.pdf (this page is also available from within R). This guide is intended for those who have never used R. As a minimum, you should read (and work through) to the end of the next section.

An Introductory Session

This section describes a session on the University IT network.

After logging in, click on **Start** (the round windows icon in the bottom left corner of your screen), then **All Programs**, then choose **Statistics**, **R**, and finally **R i386 3.5.1** . This will open a large window called "R Gui" (GUI stands for Graphical User Interface). This large window can contain one or more subwindows, in particular:

- An R Console window (which should always be present). This is where you can type your commands.
- A graphics window will appear automatically when any plotting commands are issued.
- An R Information or Help window will appear when you ask for information about particular commands.
- An R editor (a bit like NOTEPAD) where you can type commands to be saved, or run.
- Note that the choices in the R Gui menu bar vary, depending on which subwindow is highlighted (active).

Click the mouse in the console window to ensure the window is "active". One of R's features is its graphics capabilities. After the graphics window first appears, you can tidy up the window display by clicking on **Windows** → **Tile** on the R Gui menu bar.

You can keep a history of your plots by making the Graphics window active, and then clicking on **History** → **Recording**. Previous plots can then be viewed by using the PgUp, PgDn keys (when the Graphics window is active).

Overleaf is a simple annotated list of commands, which you should try out yourself for practice. Note the # precedes a comment (and need not be typed).

```
x = 1:10                          # x is a vector of integers 1 to 10
                                  # the assignment operator is a = (alternatively <-).
x                                 # list the contents of (the vector) x
str(x)                            # displays the structure of any R object
mean(x)                           # the answer 5.5 should appear
var(x)                            # the answer 9.166667 should appear
x[c(2,3,4)]                       # the subvector of 3 elements of x
errors = rnorm(10,0,2)            # a vector of 10 normal N(0,4) random numbers
bill = 5 + 3*x + errors           # a vector of simulated data
                                  # note that this uses vector arithmetic
plot(x, bill, bg="white")         # a scatter plot of the simulated data
lm1=lm(bill~x)                    # fit a straight line to the data
abline(lm1)                       # add the line to the plot
z = sqrt(bill)                    # take square roots of each element of bill
plot(x,z)                         # a new plot
ls()                              # lists all the variables in your "workspace"
rm(list = ls())                   # removes (deletes) (almost) all variables
for (i in 1:3){print(sd(x[-i]))}  # a simple "for" loop, in which the standard
                                  # deviation is printed for the data x_1,...,x_n
                                  # with the ith observation omitted (i = 1,2,3)
```

Reading in data from files. Data can be read from the web or the computer filesystem. For example
`snowfall=scan("http://www1.maths.leeds.ac.uk/statistics/R/buffalo.txt")`
or from the IT file system (including the hard drive). Similarly, you can read from a file which contains R commands – either on the web, or on the filesystem. For example
`source("http://www1.maths.leeds.ac.uk/statistics/R/quakes.R",echo=T)`.

Spaces. In general R is tolerant of extra spaces inserted in commands to improve readability (e.g. `x=2` is the same as `x = 2`. The main exception where spaces should not be used is:

- within operators; e.g. `<-` and `%*%` not `< -` and `% * %`

Upper and lower case matters: `BILL` is different from `bill` is different from `Bill`.

Mistakes If you make a typing error, the easiest way to fix it, is to use the UP arrow key to retrieve the previous command(s), then use the left/right arrow keys to make changes, and finally hit RETURN. (*Note that you cannot use the mouse in this process.*) Common mistakes include: missing brackets or commas (`Error: syntax error`) and mistyped function names (`Error in ...: couldn't find function "mistiped"`). If the syntax is so far correct then the next prompt may be a `+` in which case the command is incomplete. You can then type the remaining characters.

Using the Help System

R contains hundreds of functions to carry out various statistical and numerical tasks. Click on **Help → Html help** to start up a help window in Explorer.

In Explorer either click on **Search Engine & Keywords** or to browse click on **Packages** which will bring up a list of packages. Most commonly used commands are part of **base**. Note that most help pages have examples (at the end) and references. *The examples are extremely useful to understanding a command. You can copy and paste these into your console window to gain further understanding.*

If you know (or can guess) the name of a command then type (for example):

```
help(mean)                       # or ?mean
```

To see (and run) just the examples, type (for example):

```
example(mean)
```

Working in R

There are two main ways to work with R.

- Using the command line to type your commands, one line at a time, in the Console window.
- Using the R editor window to create and run a larger set of commands. *This is the recommended procedure for most assignments as it makes it easier for record-keeping, corrections and modifications.*

(i) From the menu bar click on **File** → **New Script** (to open an empty Editor window), or **File** → **Open Script** (to open an Editor window with an existing file of commands).

(ii) Type in the commands (suitably annotated). This window is similar to NOTEPAD in its editing capability.

(iii) Ensure the editor window is active. Then click on **Edit** → **Run All** to execute all the commands, or use the mouse to select a subset of commands and click on **Edit** → **Run Selection**. [Note the short-cuts, and use them if you prefer.]

(iv) To save your commands in a file for later use, ensure the editor window is active and click on **File** → **Save** (for an existing file), or **File** → **Save as** (to create a new file).

Note: Typing `source('filename.R')` in the command line is almost synonymous with running this file of commands from the R editor. The only difference is that paste does not echo the output in the Console window.

Printing Work

The best way to print your output is to copy-and-paste from R into WORD and to print from WORD. Both text and graphics can be copied. It is best not to try to print directly from R.

To copy text, make the R Console window active. Highlight the text you want to transfer and from the menu bar click **Edit** → **Copy**. Then make the WORD window active, position the cursor where you want the text to be inserted, and from the WORD menu bar click **Edit** → **Paste**. You should then change any R output to be in a typewriter-style font, eg Courier, to ensure that the proper formatting is preserved and distinguish it from the rest of your document.

To copy a graph, make the R Graphics window active, and from the menu bar click on **File** → **Copy to the clipboard** → **As a Metafile** (or <CONTROL>-w). In WORD, position the cursor where you want the graph to be inserted, typically on an empty line. From the menu bar click on **Edit** → **Paste** (or <CONTROL>-v).

It is possible to resize pictures.

1. Position the mouse arrow over the graph and click once. A box should appear around the graph with 8 little black squares, *control points.*
2. Position the mouse arrow over one of these control points and, holding down the left mouse button, move this square to the desired new position. Repeat until satisfied.

Saving Work and Exiting from R

To terminate the session, make the R Console window active, then click on **File** → **Exit** (or type q() in the R Console window). You will be prompted to save the "workspace image". If you respond "yes" then the workspace will be given a standard name, and be reloaded automatically the next time you start R. This will save (*only*) your variables, and not the commands that were used to obtain them, nor the plots that you obtained. For example, for the suggested commands on page 2, you will save the data contained in `bill, x, errors, z`. In general, there are four things you can save: data, commands, the session, and individual plots.

- To save your data, make active the R Console window, click on **File** → **Save Workspace** and give a filename when prompted (of type "`.RData`"). When you next log on, click on **File** → **Load Workspace** and give the file name to recover your workspace.
- In most cases, programs take longer to write (or even to type) than they do to run. So it is usually preferable to save the commands in a file, and then to re-execute them as required (see the section above "Working in R"). You can save all your commands by clicking on **File** → **Save History**, but you will probably need to edit this file to at least remove the typing errors.
- Individual plots can be saved (in various formats) for later use. Simply click on the plot, and select the format of choice.
- To save *everything* which has appeared in the R Console window, you can use **File** → **Save to a File**. This will save your commands (including any which were mistyped), and all the output, but not the data. It can require a lot of editing before it is useful.

More Advanced Features of R

Among other things R has: an effective data handling and storage facility; a suite of operators for calculations on arrays (in particular matrices); a large integrated collection of intermediate tools for data analysis; graphical facilities for data analysis and display either directly at the computer or on hardcopy; and a well developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.

Note that R is not "menu-driven", and getting used to the command line approach may take some getting used to. However, this leads to much more flexibility in practice.

There is an important difference in philosophy between R and other main statistical programs. In R a statistical analysis is normally done as a series of steps, with intermediate steps being stored as objects. For example, whereas SAS or SPSS will give copious output from a regression analysis. R will give minimal output and store the analysis in a fitted object for subsequent interrogation by further R functions.

Installing R on your PC

- These instructions assume you are using MS Windows; for Macs and Linux select the appropriate option when you get to step 2 below. If your PC is connected (with a reasonably fast connection) to the internet:

1. Open the web page: cran.r-project.org/
2. Click on `Download R for Windows`
3. Click on `install R for the first time`
4. Click on `Download R 3.5.1 for Windows` and save the file `R-3.5.1-win.exe` to your computer.
5. Locate the file `R-3.5.1-win.exe` on your computer and run it by double-clicking on it. This will then start to install R to your PC.
6. At the end of the procedure, you should have an R icon on your desktop.

- If you have no (or slow) internet connection:

1. Open the web page listed above and locate the relevant files.
2. Copy the files to a USB memory stick or similar device
3. Go home and install R from your USB stick